# NAVY Topic 113: Object Recognition Chip

A high-speed object recognition chip based on a biologically-realistic Hybrid Temporal Processing Element

Progress Report # 1

Intelligent Reasoning Systems, Inc.
P.O.Box 30001, Dept. 3ARP
Las Cruces, NM 88003

Wednesday 26 May 1993

This report traces the progress made by Intelligent Reasoning Systems, Inc. (IRSI) on contract N00014-93-C-0118, from April 1, 1993 to May 31, 1993. The structure of this report is as follows: introduction and Phase I objectives, progress to date, projected progress, and appendices containing example programs.

**Technical Objectives:**

The technical objective of this Phase I effort is to evaluate the feasibility of a binocular object recognition system for the integrated active vision system (AVS) currently under development at IRSI. The AVS based on a custom very large scale integration (VLSI) Hybrid Temporal Processing Element (HTPE) developed by IRSI (patent pending) [1-4].

Active vision implies that the image collection and processing operations form a single, integrated, adaptive system in which high-level processes, such as object recognition, can augment lower-level processes through feedback connections and can itself be controlled by higher levels. For example, the object recognition process can instruct the oscillating saccadic perception system to enhance the image resolution by a particular amount to assist in the object recognition process. Higher level perception systems can guide the object recognition processes by using expectations of what objects might be present in the image. Therefore, the AVS behaves a single image collection and processing system with coupled low- and high-level capabilities.

The IRSI AVS incorporates principles derived from experimental analysis of mammalian visual systems. Mammalian visual systems employ multiple concurrent, cross-coupled processing pathways to simultaneously infer location, motion, shape, color, and texture from images of an object obtained asynchronously and from multiple points of view (e.g. [5-6]). Processing occurs asynchronously, with continuous feedback between early and later processes. For example, low-resolution location and motion detection guide eye movement and visual attention through a fast feedback loop, while also contributing to slower, higher-resolution downstream motion, shape, and texture processing.

The AVS design is based on an asynchronous analog encoding of location and motion data, and the custom VLSI HTPE. The HTPE closely models the generic spiking neurons of the human brain [1-4]. Spiking neurons perform time-based processing using impulse-like Action Potentials (APs) and slowly time-varying Post Synaptic Potentials (PSPs). HTPEs can be used to encode and process information in terms of amplitude, frequency, phase, and time delay. In HTPE systems transient and steady-state waveforms are used to provide real-time processing capabilities. The HTPE can operate on analog data at frequencies in excess of 100 MHz, allowing rapid oversampling methods to be used for resolution enhancement, motion detection, and multiple template matching. HTPEs have low device count and power dissipation, and can be fabricated in small layout areas. The AVS is intended for eventual on-board application in robots, intelligent machine tools, and other autonomous sensory-motor systems that require visible, IR, UV, or similar input. The basic architecture developed for the AVS will, however, be useful for any sensory input that can be represented in a plane, and for which spatial continuity is a relevant constraint. The current AVS design, shown in block form in Fig. 1 incorporates asynchronous data collection and processing, multiple processing streams, and attentional feedback.

The initial stage of the AVS collects data asynchronously from the primary sensor, and provides a parallel analog encoding. This encoding allows early preprocessing circuitry for tasks such as low-level resolution enhancement and motion-detection to be built into the retina, and to operate on data from each pixel or neighborhood in parallel. The preprocessed data is then encoded in asynchronous pulse streams and mapped, again in parallel, to hierarchical feature detection systems in which neighborhood relations between processing elements corresponding to pixels or pixel neighborhoods are preserved. The processing layers in this initial AVS stage correspond to the early processing stages in the retina of the human visual system (HVS). The initial stage of the AVS, from here on referred to as the integrated saccadic perception image resolution enhancement system (INSPIRES), was developed under an Army Phase I SBIR effort and is currently in the Phase II review process.

Phase I proposals for the development of the low-level feature extraction systems of the AVS were also submitted, but were not funded. These systems are vital to the current project, and therefore, must also be developed. The preliminary development of these systems was performed under the Army INSPIRES project and will be further developed over the course of this project. Since the feature extraction systems provide direct input to the binocular object recognition system under development in this Phase I effort, their concurrent design will allow them to be tailored so as to best compliment each other.

The specific objectives of this Phase I effort are:

1. The AVS primary feature detectors will process data from each pixel neighborhood in parallel. Fusing data from two primary feature detectors raises two hardware issues: a) data will probably have to be multiplexed off of the feature-detector chips and on to the fusion chip; b) multiple fusion chips may be needed, raising the issue of preserving spatial coherence between overlapping fields. Alternative strategies for addressing both of these issues need to be evaluated.

2. The primary depth cue is binocular disparity. Disparity between the primary feature maps is sensitively dependent on the resolutions of both the retinas themselves and the downstream processing. The mapping of disparity to depth for various realistic combinations of retinal spacing, retinal resolution, and feature resolution needs to be investigated.

3. The binocular fusion circuitry for calculating 3d location and motion of objects and features needs to be designed and implemented. An appropriate encoding for combined object and motion data needs to be developed. The fusion system design needs to be general enough to work for a variety of depth-to-disparity mappings.

4. Analog logic needs to be developed for combining primary features into simple shapes as the first step in shape recognition. Conventional artificial neural networks (ANNs) can classify observed shapes into learned classes, but require high connectivity if many classes are used. A hierarchical classification system for constructing complex shapes from simple ones may be more practical than a conventional ANN for hardware

St-A per telecon, Dr. Davis, ONR/Code
1142CN. Arl., VA 22217

6-4-93 JK

implementation. The relevant tradeoffs between these two approaches need to be identified and evaluated.

5. Occlusion is an excellent depth cue, but can only be used if occluding and occluded objects can be distinguished. The ability of the shape-classification system for distinguishing occluded objects needs to be tested.

6. In general, the optimal split between object recognition for attention and targeting and shape recognition needs to be identified, as do the points at which feedback from higher-level processing can be used to disambiguate objects and shapes.

## Progress to Date:

*AVS Structure*

Current advances in MCM technology now allow for three-dimensional IC implementations [7-12]. One such process referred to as "cubing" [12] stacks as many as 100 thinned ICs (0.2 mm thick) into a solid block. This technique greatly increases circuit density and thermal conductivity while improving reliability by eliminating most of the interconnect junctions such as wire bonds, TAB pads, and bump bonds. Shorter signal lines mean lower line capacitances, which map directly to decreased drive requirements and increase processing speeds. Due to the three-dimensional IC advances and the fully parallel nature of the HVS from which we draw motivation, the IRSI AVS is designed as a fully parallel image collection and processing system.

*Feature Extraction*

The role of the feature extraction system in the AVS is to extract low-level spatial and temporal features, such as, lines and edges, moving lines and edges, and accelerating lines and edges. The structure and functionality of the feature extraction system draws it motivation from a novel class of information-processing systems called *cellular neural networks* (CNNs) are currently of major interest in image processing applications [13-17]. Like a neural network, a CNN is a large-scale non-linear analog circuit which processes signals in real time. Like *cellular automata* (CA), CNNs are composed of a massive aggregate of regularly spaced circuit clones, called cells, which communicate directly with only their nearest neighbors; no global connections exist. Global information is obtained indirectly because of the propagation effects of the continuous-time dynamics of CNNs. The real-time processing capabilities and local connection scheme (ideal for VLSI implementation) make CNNs attractive for high-speed image processing applications.

In a conventional implementation of a CNN each cell is connected to its nine nearest neighbors (including a self feedback connection) by linear weights (positive or negative links). Other nearest-neighbor connection schemes are possible including three, four, and six links. The dynamics of the CNN are determined by the nearest neighbor connections. These coefficients (connection weights) can be specified in a square array which is referred to as a *cloning template* . In conventional digital image processing neighborhood filters (called *kernels* ) resembling cloning templates are serially swept across an image. In CNNs however, the filters are applied in parallel, which removes the speed bottleneck that plagues digital systems and results in real-time processing capabilities. To serially pass a 3 X 3 pixel filter over a standard 512 X 512 pixel image once would require over two million multiplications and additions. To further complicate matters, when image filtering is implemented discretely in a digital system the propagation effects of continuous dynamics are not maintained and, therefore, large kernels (maybe 100 X 100 pixels) are needed to obtain global information. This pushes the number of required digital operations to impossible limits (on the order of 1 billion operations per frame).

A C-based software package has been developed by IRSI for the simulation of conventional CNNs to examine the effects of various cloning templates and output functions. The simulator (included as Appendix A) reads a simulation deck (included as Appendix B) and executes the series of specified instructions. The instructions specify input and output image filenames, cloning template filenames, output function filenames, and network initial conditions. The simulation deck is divided into cycles, each of which specifies a particular CNN routine. For example, a simulation deck may contain two cycles to be executed; the first cycle may be to apply a hole-filling template to a given input image and the second cycle may be to apply an edge-extraction template to the output image produced by the first cycle. The simulator sequentially executes all specified cycles until the EOF marker is found. The simulation deck approach allows for the programming of CNNs (a sequence of routines can be specified without the need for the user to reconfigure the network after each routine). In conventional CNNs the cloning templates are typically linear, time-independent coefficient matrices; in our simulator however, the cloning templates are functions, which allows non-linear, time-varying, and even logic templates to be employed. The results of each cycle are presented to the user through a custom graphical-user-interface also developed by IRSI. A demo copy of the simulator, example simulation decks, and a detailed users manual will be provided with the final report.

Examples which demonstrate the results of applying various cloning templates on given input images are shown in Figs. 2, 3, 4, and 5. Figure 2 shows the effect of applying a vertical edge-detection template to an input image (top). The template not only extracts the vertical edges present in the input image it also fills in the single pixel gaps in those edges (bottom). Figure 3 shows the effect of applying a horizontal edge-detection template to an input image (top). This template behaves similar to the vertical edge detection template in that it extracts edges and fills in single pixel gaps (bottom). Figure 4 shows the effect of applying a general purpose edge-detection template to an input image (top). The template extracts all concave and convex edges in the image, but does not fill in any single pixel gaps (bottom). Figure 5 shows the effect of applying a hole-filling template to an input image (top). The template fills in the holes in the input image and smoothes the outer edge of the object (bottom).

Templates or series of templates that perform higher-order operations such as, simple motion detection, centroid estimation, estimation of two-dimensional object displacement, object segmentation, and object recognition are currently under development. Conventional CNNs are fairly well suited for spatial feature extraction and can be made to perform some simple temporal feature extraction, but their overall utility is limited by their passive settling behavior and time-independent processing. A more powerful form of locally-connected network can be found in the HVS.

Conventional CNN architecture's are directly derived from biological networks. In fact, locally connected networks are the dominant structure in the HVS, from the low-level center-surround networks of the retina to the higher-level "simple" and "complex" cells of the primary visual cortex. The simple and complex cells of the HVS even though similar to CNNs are capable of more sophisticated behaviors. We will now detail the simple and complex cells of the primary visual cortex and HTPE-based networks that possess similar functionality. The exact electrophysiology of simple and complex cells is not known but their high-level functionality is actually quite well understood. Since it is this functionality we are interested in we can use the cues of the HVS to develop a diverse set of spatiotemporal HTPE-based processors with similar capabilities.

*Simple cells*

Cortical simple cells are similar to retinal ganglion cells, lateral geniculate cells, and circularly symmetric cortical cells in that they have a small, clearly delineated receptive field [18]. A small spot of light incident upon the field produces either an on or off response, depending on where in the field it falls. The difference between the simple cells and the cells at earlier stages is the geometry of the excitation and inhibition maps. The maps of the earlier cells are circularly symmetric (a circular region of excitation or inhibition is completely surrounded on all sides by a region of the opposite "polarity"). Cortical simple cells are more complex. The inhibition and excitation regions are always separated by a straight line or two parallel lines (see Figs. 6a, 6c, and 6g). The most common configuration is that of Fig. 6a in which a long narrow region of excitation is flanked on both sides by larger regions of inhibition. The most potent stimulation for this type of cell is a long narrow slit of light that is properly positioned and oriented as to cover the excitatory portion of the receptive field. Even the slightest misalignment or misorientation will cause the response to decline. Simple cells are found at all orientations and visual field positions.

Figures 6b, 6d, and 6h show possible methods for the implementation of the various simple cell geometries. The simple cell of Fig. 6a can be constructed as shown in Fig. 6b. On center ganglion cells whose centers overlap along a line at a given orientation provide excitatory input to the simple cell. A small spot of light that excites a few of the ganglion cells will cause a slight response in the simple cell. A line of light at the proper orientation, however, will excite many ganglion cells producing a large response in the simple cell. The HTPE implementation of this simple cell geometry is shown in Fig. 6e and its corresponding response to a spot of light (top) and a line of light (bottom) is shown in Fig. 6f. The simple cell geometries of Figs. 6c and 6g can be similarly implemented in HTPE networks.

*Complex Cells*

Complex cells, like simple cells, are sensitive to lines at particular position and orientation, but unlike simple cells, complex cells only produce a large sustained response if the line is moving in a particular direction [18]. Simple cells produce a maximal response to stationary lines; a moving line elicits a response only during the brief time it occupies the excitatory region of the cell's receptive field. In complex cells a properly position and oriented moving line produces a large response from the time it enters the receptive field until the time it exits the field. Some complex cells do respond to stationary lines but the position of the line in the field is unimportant; only the orientation is relevant. Figure 7a shows examples of lines at various orientations with there relative movement indicated by the arrows. An HTPE implementation of a directionally sensitive set of complex cells is shown in Fig. 7b (the cross-hatched rectangles represent -+- simple cells). Each simple cell feeds excitation (white triangle) into a complex cell and inhibition (black triangle) to a neighboring complex cell. If the right-most simple cell becomes active (signifying a properly positioned and oriented line of light is incident upon its receptive field) it excites it corresponding complex cell and inhibits the complex cell to its immediate left. If the line of light moves to the left and excites the next simple cell, the simple cell will try to excite its complex cell, but due to the lateral inhibition already applied to that complex cell it will remain inactive. Therefore, a line of light moving in such a way as to stimulate the simple cells from right to left will not cause the output of the complex cells to become active. A line of light moving in the opposite direction, however, will cause the complex cells to become active because there are no inhibitory links in that direction. Figure 7c shows several simulated behaviors of the set of complex cells shown in Fig. 7b. The top plot in Fig. 7c shows the response of the complex cell system to a line of light moving left to right (only the first cell fires because its inhibitory input is not active). The center plot in Fig. 7c shows the response of the complex cell system to a line of light moving right to left (all cells fire because there are no right to left inhibitory cross-links). The bottom plot in Fig. 7c shows the response of the complex cell system to a line of light moving left to right at a rate slow enough to be unaffected by the inhibitory cross-links (due to the finite length of the inhibition a slow moving line can pass through). Below we describe how a HTPE complex cell network can be made sensitive or insensitive to a particular rate of motion.

Figure 7d shows a universal complex cell connection scheme that allows the directional sensitivity of the system to be electronically programmed. Components (excitatory and inhibitory cross-links) not being used at a particular time can be electronically disabled. The HTPE-based universal complex cell has capabilities beyond that of the biological complex cell, for example 1) the directional sensitivity is programmable, 2) the network can be made sensitive to velocity, and 3) the network can be made sensitive to acceleration. The reason for this versatility can be understood by examining Fig. 8a, which shows an illustration of the excitatory (top) and inhibitory (center) post synaptic potentials (EPSP and IPSP, respectively) of the HTPE. These waveforms are characterized by their amplitude, delay, and duration, which are all independently controllable. The amplitude of the waveform is its height above or below the resting potential of HTPE. The delay of the waveform is the time it takes for the waveform to appear after an action potential (AP) has fired it. The duration is simply the time extent of the waveform after it has appeared. Figure 8a also shows the waveform (bottom) produced by summing the EPSP and IPSP. Figure 8b is a simulation of a single HTPE with one EPSP and one IPSP input. The voltage "v1.soma" is similar to the waveform of Fig. 8a (bottom). Notice the HTPE begins to fire APs as soon as the EPSP arrives and the firing rate increases once the IPSP has elapsed. These time-dependent waveforms are what enable HTPE-based complex cells to have such diverse behaviors.

To make a complex cell velocity dependent can be achieved as follows. The excitatory inputs to a complex cell from its corresponding simple cell and its neighboring simple cell can be made of subthreshold amplitude. This means that in order for the complex cell to become active both excitatory input must be active at the same point in time. By setting the delay and duration of the excitatory input from the neighboring simple cell to a given value it will be coincident with the other excitatory input if the input line of light is move at an acceptable speed. Since the simple cell's excitatory input is not strong enough by itself to cause the complex cells to become active the cross-link excitation can be used to produce directional selectivity as well as velocity sensitivity (since velocity is speed and direction directional sensitivity is implied). To clarify this behavior consider the following three examples:

1) If the excitatory input from the a simple cell to its left neighbor complex cell is set to have a ten nanosecond delay and a twenty nanosecond duration the complex cells will become active if a properly positioned and oriented line of light is moving across the receptive field at the proper rate as determined by

$$\frac{\text{distance}}{10\text{ns}} \leq \text{rate} \leq \frac{\text{distance}}{20\text{ns}}$$

where, distance is dependent on the dimension of the simple cell receptive fields and the characteristic of the imaging system optics, such as system focal length. The behavior of this universal complex cell system is a velocity bandpass filter for straight lines. If the delay and duration parameters are not uniform across the system it also becomes acceleration sensitive.

2) If the right to left inhibitory cross-links are used (with the same delay and duration settings as the excitatory cross-links of example 1) instead of the excitatory and the direct simple cell to complex cell excitatory link is made strong enough to cause activation by itself, the network will behave as a velocity bandstop straight line filter. The stop band is the same as the pass band of example 1. The non-uniform connection scheme can also be employed in this configuration to yield acceleration sensitivity.

3) Cross-links (excitatory and/or inhibitory) can be made in both directions allowing for different velocity and acceleration criteria depending on the direction of movement. This will allow extremely complicated spatiotemporal processing to be achieved with a single complex cell network.

One additional behavior of complex cells is known as "end stopping". End stopped cells are not only position, orientation, and directionally sensitive, but they are also sensitive to the length of the incident line. Lines of the ideal length cause maximal complex cell output, whereas lines shorter or longer than the ideal length cause only moderate complex cell output. One possible method for implementing the behavior of an end-stopping complex cell is to combine the output of multiple simple cells, as shown in Fig. 9a. Three -+- simple cells with their positive regions lying adjacent along a common line provide input to an end-stopping complex cell. The two simple cells on the ends provide inhibitory input and the center simple cell provides excitatory input to the end-stopping complex cell. Figure 9b shows the simulated behavior of an end-stopping complex cell. The cell output becomes active when the excitatory portion of simple cell B is stimulated by a short line of light. As the line length begins to expand the complex cell output begins to decrease and then stops when the line is long enough to fully stimulate simple cells A and C.

Since simple cells are sensitive to lines at particular orientation they are analogous to conventional CNNs running edge extraction cloning templates. A major difference between simple cells and CNNs is that a CNN extracts the edges in a image on a frame by frame basis. A frame is applied to the CNN and as the network settles the edges appear on the output. The next frame is then applied and the process repeats. The output of the last cycle has no direct effect on the operation during the current cycle. Conventional CNNs and neural networks for that matter are memoryless systems unless previous cycle information is explicitly stored and reentered as input during future cycles or multi-layer structures with feedback are employed. On the other hand HTPE-based simple cells are inherently time-based systems in which previous system activity directly affects the current and future behavior of the system. The input image is continuously processed; edge information passes through the simple cells to higher-level processes and changes in location or orientation are continuously detected and acted upon.

Complex cells have no direct CNN analogy due to their time-dependent nature. In conventional CNNs the determination of arbitrary object motion between frames requires the serial application of ten or more cloning templates. This limitation alone prevents CNNs from continually processing image motion. Velocity and acceleration sensitivity are an even less practical objective to achieve in conventional CNNs. The time-dependent processing of biological neurons make them ideal for the implementation of real-time continuous image processing systems.

## Projected Progress:

Due to the fully parallel architecture adopted for the AVS, multiplexing methods for the transport of data off one chip and onto another are not a major issue at this point. Since subsequent processing stages in the AVS may require data multiplexing, however, we have developed several methods for encoding multiple pieces of data onto a single asynchronous pulse stream. Methods for extracting the individual pieces of data from the pulse stream have also been developed. The location of the data piece in the pulse stream is used to maintain spatial coherence. These data encoding and decoding systems will be further refined during the next work period. System schematics and simulation results will be presented in the second progress report.

The next step in the visual processing of the AVS is the determination of deptn. Multiple mechanisms are used in the HVS for depth perception, some of which utilize the data provided by a single eye, while others utilize the information provided by both eyes. Mechanisms that use the information provided by a single eye include: occlusion, parallax, rotation of objects, relative size, shadow casting, and perspective [18].

Occlusion - When an object is in front of another object and partially blocks its view we judge the front object to be closer and the back object to be farther away.

Parallax - The relative motion of near and far objects that is produced when we move our heads from side to side or up and down is a powerful depth cue. The larger displacement of an object in the visual field implies that it is closer than an object with a smaller displacement.

Rotation of objects - The rotation of a solid object by even a slight amount can make its shape immediately apparent. The object shape or change in shape can therefore be used to judge depth.

Relative size - When the size of an object is roughly known we can judge it distance by comparing its size in the image to its expected size.

Shadow casting - A bump on the wall that juts out is brighter on top if the light source comes from above, and a pit in the surface lit from above is darker in its upper part. This depth information is useful for texture analysis.

Perspective - The image of parallel lines, like railroad tracks, as they go off into the distance draw closer together. This is an example of perspective, which provides powerful depth cues.

Each of these mechanisms can be used to provide depth information individually or in combination. Since each mechanism has it limitations the combined use of all of them will provide more accurate depth determination.

Some of the above depth determination mechanisms rely on expectations and knowledge of the surrounding environment. Expectation-guided perception and knowledge representation systems for the AVS are currently under development in other projects. These systems will provide high-level guidance to the object recognition system an other lower-level AVS systems. This guidance will assist the object recognition system in the tasks of recognition and depth determination. The interface between the high-level systems and the object recognition system will be developed during the next work period as will the methods for implementing the various depth determination mechanisms (this will include the design and simulation of the required HTPE networks).

Perhaps the most important mechanism for assessing depth is *stereopsis*, which is the fusing of the information provided by both eyes. Stereopsis is performed in the primary visual cortex by a more sophisticated form of complex cell. These complex cells have the same properties as the earlier mention complex cells - line and edge sensitivity, movement sensitivity, directional sensitivity, and end-stopping - plus additional properties relevant to stereopsis. To keep things clear we will refer to the new form of complex cells as fusion cells.

Fusion cells fall into four basic categories: disparity insensitive cells, tuned excitatory cells, near cells, and far cells. The latter three being collectively referred to as disparity tuned cells. To discuss the properties of the fusion cells we must first introduce the concept of disparity. The following definition of *disparity* is taken directly from [18].

> "Suppose an observer fixes his gaze on a point P. This is equivalent to saying that he adjusts his eyes so that the images of P fall on the foveas, F (see Fig. 10a). Now suppose Q is another point in space, which appears to the observer to be the same distance away as P, and suppose QL and QR are the images of Q on the left and right retinas. Then we say that QL and QR are *corresponding points* on the two retinas. Obviously, the two foveas are corresponding points; equally obvious, from geometry, a point Q' judged by the observer to be closer to him than Q will produce two noncorresponding images Q'L and Q'R that are farther apart than they would be if they were corresponding (see Fig. 10b). If you like, they are outwardly displaced relative to each other, compared to the positions corresponding points would occupy. Similarly, a point farther from the observer will give images closer to each other (inwardly displaced) compared to corresponding points."

This lack of correspondence is referred to as disparity. This definition will now be used to describe the properties of the four categories of fusion cells.

> Disparity insensitive cells are stimulated if a slit of light is swept across both retinas simultaneously. The position of this slit of light, however, has no significant effect on the output of the cell; only the fact that both retinas were stimulated is of consequence in disparity insensitive cells.
>
> Tuned excitatory cells are stimulated maximally only when the stimuli to the two retinas falls exactly on corresponding parts of both retinas. The amount of horizontal disparity that can be tolerated before the cells response disappears is a fraction of the width of the receptive field. Therefore, the cell fires if and only if the object is roughly as far as the distance on which the eyes are fixed.
>
> Near cells fire maximally only when the object is closer than the distance on which the eyes are fixed (outwardly displaced points).
>
> Far cells fire maximally only when the object is farther than the distance on which the eyes are fixed (inwardly displaced points).

It is believed that the primary visual cortex contains disparity tuned cells that are sensitive to particular ranges of disparity. Due to the programmability of HTPE networks a particular HTPE-based disparity tuned cell can be made selective to a variable narrow disparity range or even swept through a wide range of disparity sensitivities. Additional HTPE circuitry can be used to develop disparity tuned cells that adaptively lock-on to the exact disparity between two objects, thus providing a more quantitative measure of object distance. During the next work period various depth to disparity mappings will be examined in terms of retinal spacing, retinal resolution, and feature resolution. HTPE-based versions of all four fusion cells will also be developed and simulated.

One additional task to be explored during the next work period is the design of methods and networks for the combination of primary features into simple shapes as the first step in shape and object recognition. These methods will involve the use of feedback information from the higher-level expectation-guided perception and knowledge representation systems of the AVS.

## References

[1] M. DeYong, R. Findley, and C. Fields, "The design, fabrication, and test of a new hybrid analog-digital neural processing element," IEEE Transactions on Neural Networks, Vol. 3, No. 3, 1992.

[2] M. DeYong, T. Eskridge, and C. Fields, "Temporal signal processing with high-speed hybrid analog-digital neural networks," Jour. of Analog Integrated Circuits and Signal Processing, Vol. 2. Boston: Kluwer Academic Publishers, 1992.

[3] M. DeYong, T. Eskridge, and A. Palmer, "A coupled-grid neural network retina for real-time visual processing," Proc. IEEE 35th. Midwest Symposium on Circuits and Systems, Washington, D.C., 1992.

[4] C. Fields, M. DeYong, and R. Findley, "Computational capabilities of biologically realistic analog processing elements," In: J. Delgado-Frias (Ed.) *VLSI for Artificial Intelligence and Neural Networks*. New York: Plenum, 1992.

[5] J. Maunsell and W. Newsome, "Visual processing in monkey extrastriate cortex," Annual Review of Neuroscience, Vol. 10, 1987.

[6] E. DeYoe and D. Van Essen, "Concurrent processing streams in monkey visual cortex," Trends in Neuroscience, Vol. 11, No. 5, 1988.

[7] C. Val and T. Lemoine, "3-D interconnection for ultra-dense multichip modules," IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. 13, No. 4, 1990.

[8] C. Chao, K. Scholz, J. Leibovitz, M. Cobarruviaz, and C. Chang, "Multi-layer thin-film substrates for multi-chip packaging," IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. 12, No. 2, 1989.

[9] J. Reche, "Fabrication of high density multichip modules," IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. 13, No. 3, 1990.

[10] M. Greenstein and F. Matta, "A precision vertical interconnect technology," IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. 14, No. 3, 1991.

[11] B. Foster, F. Bachner, E. Tormey, M. Occhionero, and P. White, "Advanced ceramic substrates for multichip modules with multilevel thin film interconnects," IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. 14, No. 4, 1991.

[12] "Mission Accomplished," NASA Tech Briefs, Vol. 17, No. 5, 1993.

[13] L. Chua and L. Yang, "Cellular neural networks: theory," IEEE Transactions on Circuits and Systems, Vol. 35, No. 10, 1988.

[14] L. Chua and L. Yang, "Cellular neural networks: applications," IEEE Transactions on Circuits and Systems, Vol. 35, No. 10, 1988.

[15] T. Roska, T. Boros, P. Thiran, and L. Chua, "Detecting simple motion using cellular neural networks," Proc. IEEE International Workshop on Cellular Neural Networks and Their Applications, Budapest, Hungary, 1990.

[16] T. Matsumoto, T. Yokohama, H. Suzuki, R. Furukawa, A. Oshimoto, T. Shimmi, Y. Matsushita, T. Seo, and L. Chua, "Several image processing examples by CNN," Proc. IEEE International Workshop on Cellular Neural Networks and Their Applications, Budapest, Hungary, 1990.

[17] T. Roska and L. Chua, "Cellular neural networks with nonlinear and delay-type template elements," Proc. IEEE International Workshop on Cellular Neural Networks and Their Applications, Budapest, Hungary, 1990.

[18] D.H Hubel, *Eye, Brain, and Vision* . New York: W.H. Freeman and Company, 1988.

## Appendix A

```
for (i=0;i<row;i++) {
        for (j=0;j<column ;j++) {
            float s=0:
            int numnei=0;
            A[i][j].weight=Weight[1][1];
            A[i][j].inpweit=B[1][1];
            s=s+A[i][j].out[0]*A[i][j].weight +
            A[i][j].u*A[i][j].inpweit + A[i][j].value[0];
            A[i][j].x=i;
            A[i][j].y=j;
            neicount=0;
            A[i][j].numnei=0;
            nrows=row-1;
            ncolumns=column-1;
            minx=(i-1<0)?0:i-1;
            maxx=(i+1>nrows)?nrows:i+1;
            miny=(j-1<0)?0:j-1;
            maxy=(j+1>ncolumns)?ncolumns:j+1;
            for (x=minx;x<=maxx;x++) {
                for (y=miny;y<=maxy;y++) {
                    if ((i!=x) || (j!=y)){
                        (A[i][j].neigh[neicount]).x=x;
                        (A[i][j].neigh[neicount]).y=y;

                        if (x<i) {
                            if (y<j) {
                    (A[i][j].neigh[neicount]).weight=Weight[0][0];
                    (A[i][j].neigh[neicount]).inpweit=B[0][0]; }
                            if (y==j) {
                    (A[i][j].neigh[neicount]).weight=Weight[0][1];
                    (A[i][j].neigh[neicount]).inpweit=B[0][1]; }
                            if (y>j) {
                    (A[i][j].neigh[neicount]).weight=Weight[0][2];
                    (A[i][j].neigh[neicount]).inpweit=B[0][2]; }
                            }
                        if (x==i) {
                            if (y<j) {
                    (A[i][j].neigh[neicount]).weight=Weight[1][0];
                    (A[i][j].neigh[neicount]).inpweit=B[1][0];}
                            if (y>j) {
                    (A[i][j].neigh[neicount]).weight=Weight[1][2];
                    (A[i][j].neigh[neicount]).inpweit=B[1][2]; }
                            }
                        if (x>i) {
                            if (y<j) {
                    (A[i][j].neigh[neicount]).weight=Weight[2][0];
                    (A[i][j].neigh[neicount]).inpweit=B[2][0]; }
                            if (y==j) {
                    (A[i][j].neigh[neicount]).weight=Weight[2][1];
                    (A[i][j].neigh[neicount]).inpweit=B[2][1]; }
```

```
                              if (y>j) {
                    (A[i][j].neigh[neicount]).weight=Weight[2][2];
                    (A[i][j].neigh[neicount]).inpweit=B[2][2]; }
                              }

                         inde++;

                         s=s+
                    A[x][y].out[0]*(A[i][j].neigh[neicount]).weight+
                    A[x][y].u*(A[i][j].neigh[neicount]).inpweit;
                    neicount++;
               }
}}
               A[i][j].value[1]=I+s;
               s=compar(node,s, I));
               A[i][j].out[1]=s;
               if (n>=5) {
               for (io=0;io<4;io++) {
               FIFO[i][j].fifovalue[io]=FIFO[i][j].fifovalue[io+1];
               FIFO[i][j].fifoout[io]=FIFO[i][j].fifoout[io+1];
               }
               FIFO[i][j].fifovalue[4]=A[i][j].value[1];
               FIFO[i][j].fifoout[4]=A[i][j].out[1];
               }
               else {
               FIFO[i][j].fifovalue[n]=A[i][j].value[1];
               FIFO[i][j].fifoout[n]=A[i][j].out[1];
               }
               image[i][j]=255- (int)((A[i][j].out[1]+1)/2*255);
               fprintf(f6,"%f\n", A[i][j].out[1]);
               if (nn>5) {
               if (absl(A[i][j].out[1]-
               A[i][j].out[0])<=.05) count++; }
               A[i][j].value[0]=I+s;
               A[i][j].out[0]=s;
               }
               fprintf(f6, "\n");
               }
               printf("\nDONE\n");
               printf("The Value of n is :%d\n",++n);
               fflush(f6);
               fclose(f6);
               printf("nn = %d\ncount = %d\n", nn, count);
               nn++;
          }
               while  ((answer3=='c' || answer3=='C' ) &&
          (count!=row*column));
          }
```

## Appendix B

Sample Simulation Deck Read by CNN Simulator for CNN

```
**********************************************************************
i
temhoriz
sat
images/inimage4
i
**********************************************************************
i
template
sat
out1
**********************************************************************
s
temdiff
sat
out2
out1
**********************************************************************
i
template
sat
out3
**********************************************************************
s
temdiff
sat
out4
out3
**********************************************************************
i
template
sat
out5
**********************************************************************
s
temdiff
sat
out6
out5
**********************************************************************
i
template
sat
out7
**********************************************************************
s
temdiff
sat
out8
out7
**********************************************************************
```

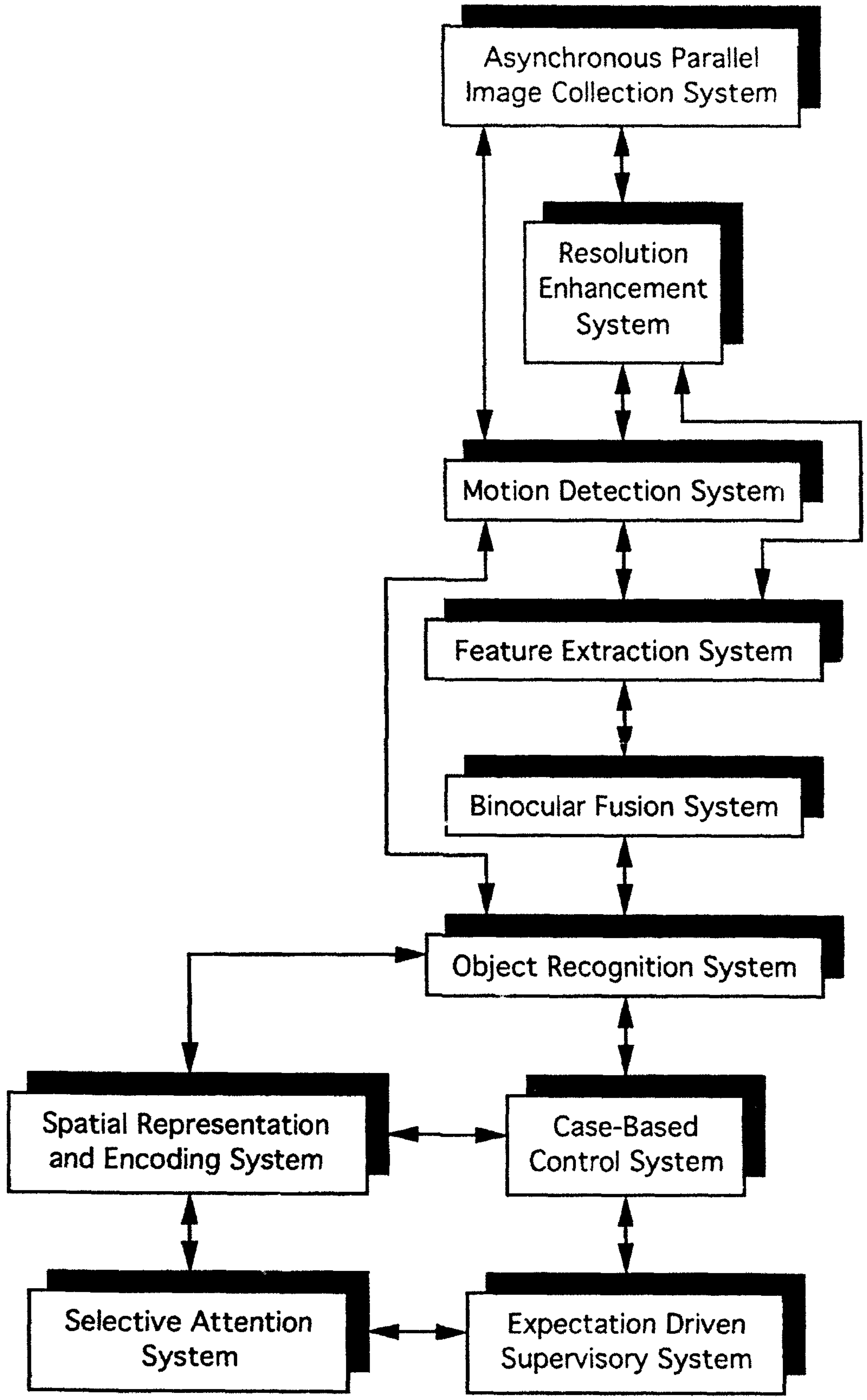

Asynchronous Parallel Image Collection System
Resolution Enhancement System
Motion Detection System
Feature Extraction System
Binocular Fusion System
Object Recognition System
Spatial Representation and Encoding System
Case-Based Control System
Selective Attention System
Expectation Driven Supervisory System

**Fig. 1:** Current IRSI AVS design, incorporating asynchronous data collection and processing, multiple processing streams, and attentional feedback.

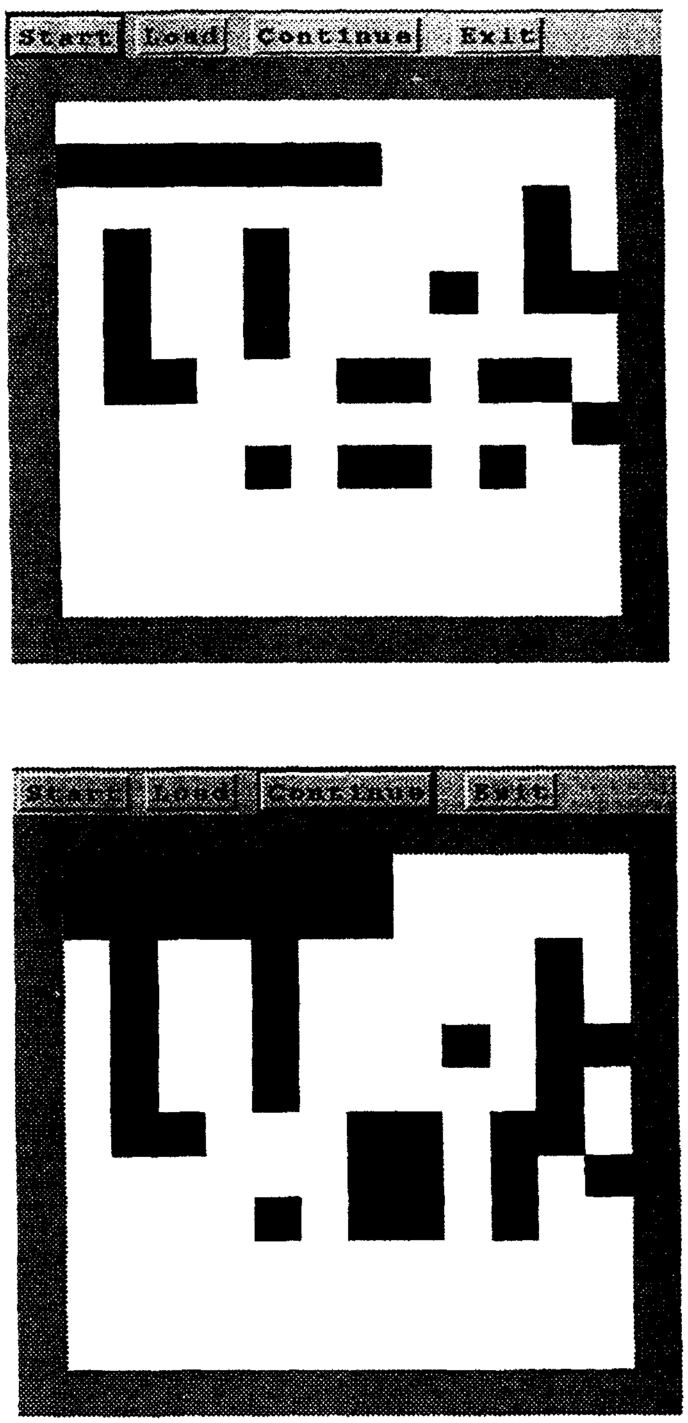
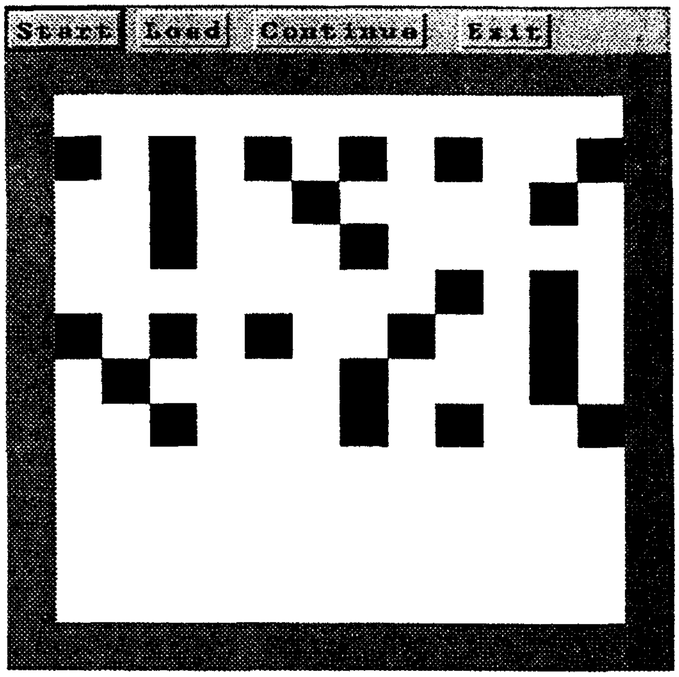


**Fig. 2:** Example of a vertical edge detector. Effect of applying a vertical edge-detection template to an input image (top). The template not only extracts the vertical edges present in the input image, it also fills in the single pixel gaps in those edges (bottom).



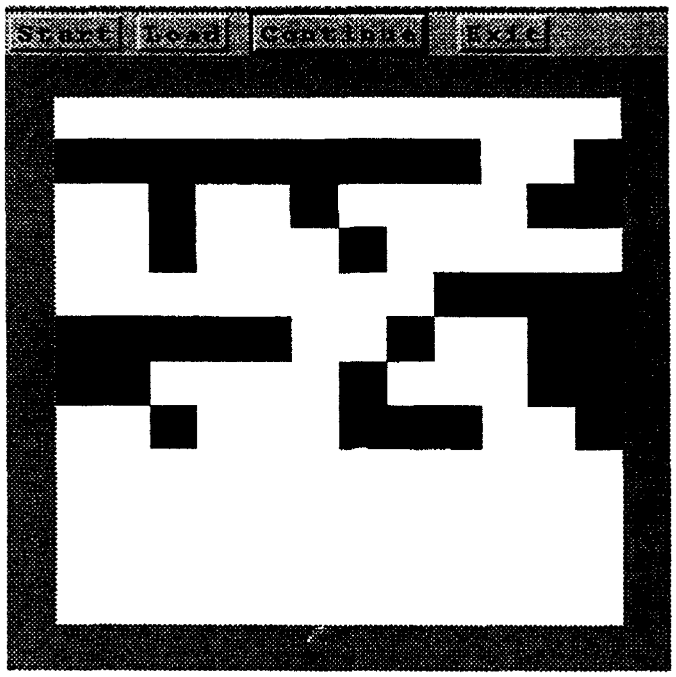
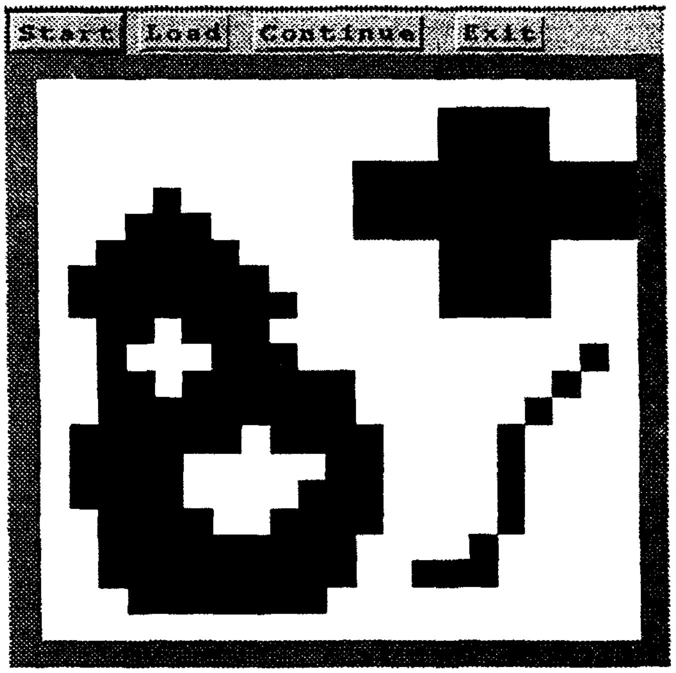


**Fig. 3:** Example of a horizontal edge detector. Effect of applying a horizontal edge-detection template to an input image (top). The template not only extracts the horizontal edges present in the input image, it also fills in the single pixel gaps in those edges (bottom).



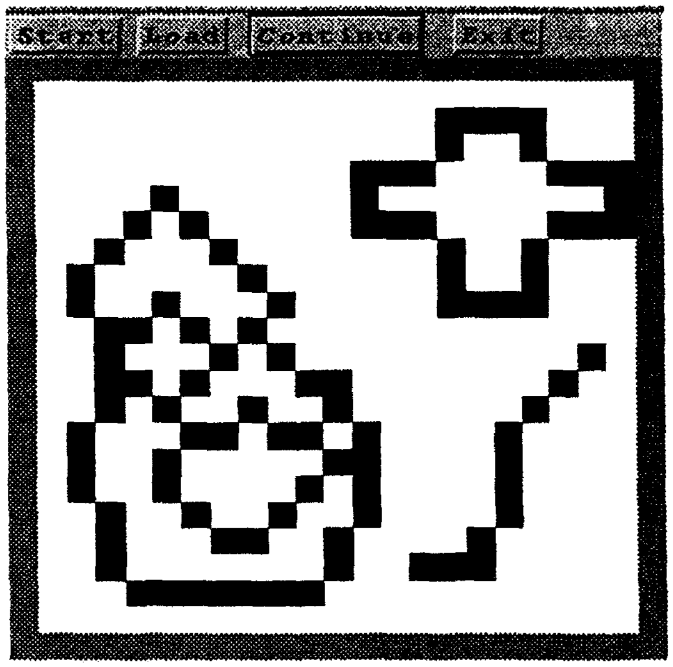
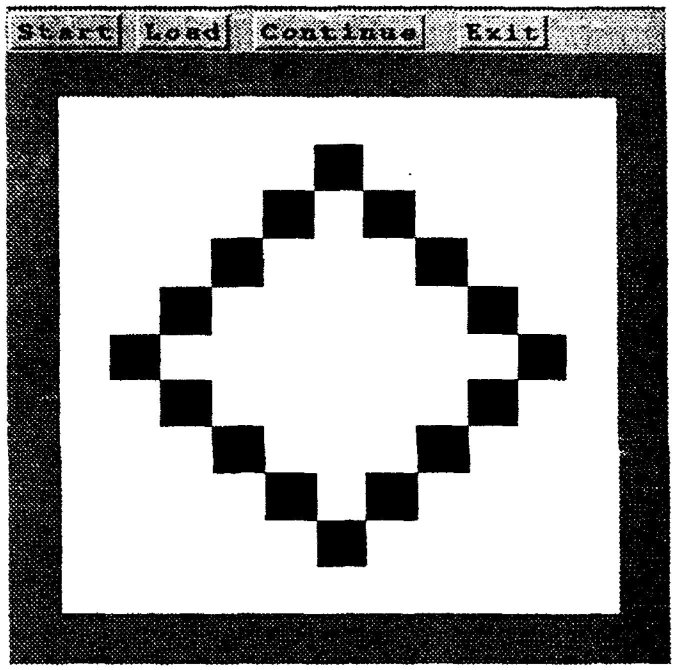


**Fig. 4:** Example of an edge detector. Effect of applying a general purpose edge-detection template to an input image (top). The template extracts all concave and convex edges in the input image but does not fill in any single pixel gaps in those edges (bottom).



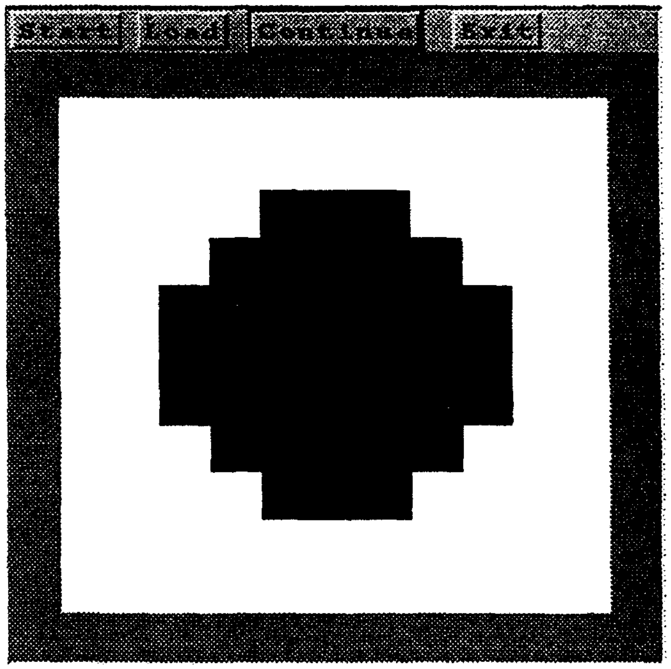


**Fig. 5:** Example of a hole filler. Effect of applying a hole-filling template to an input image (top). The template fills in the holes in the input image and smoothes the outer edge of the object (bottom).

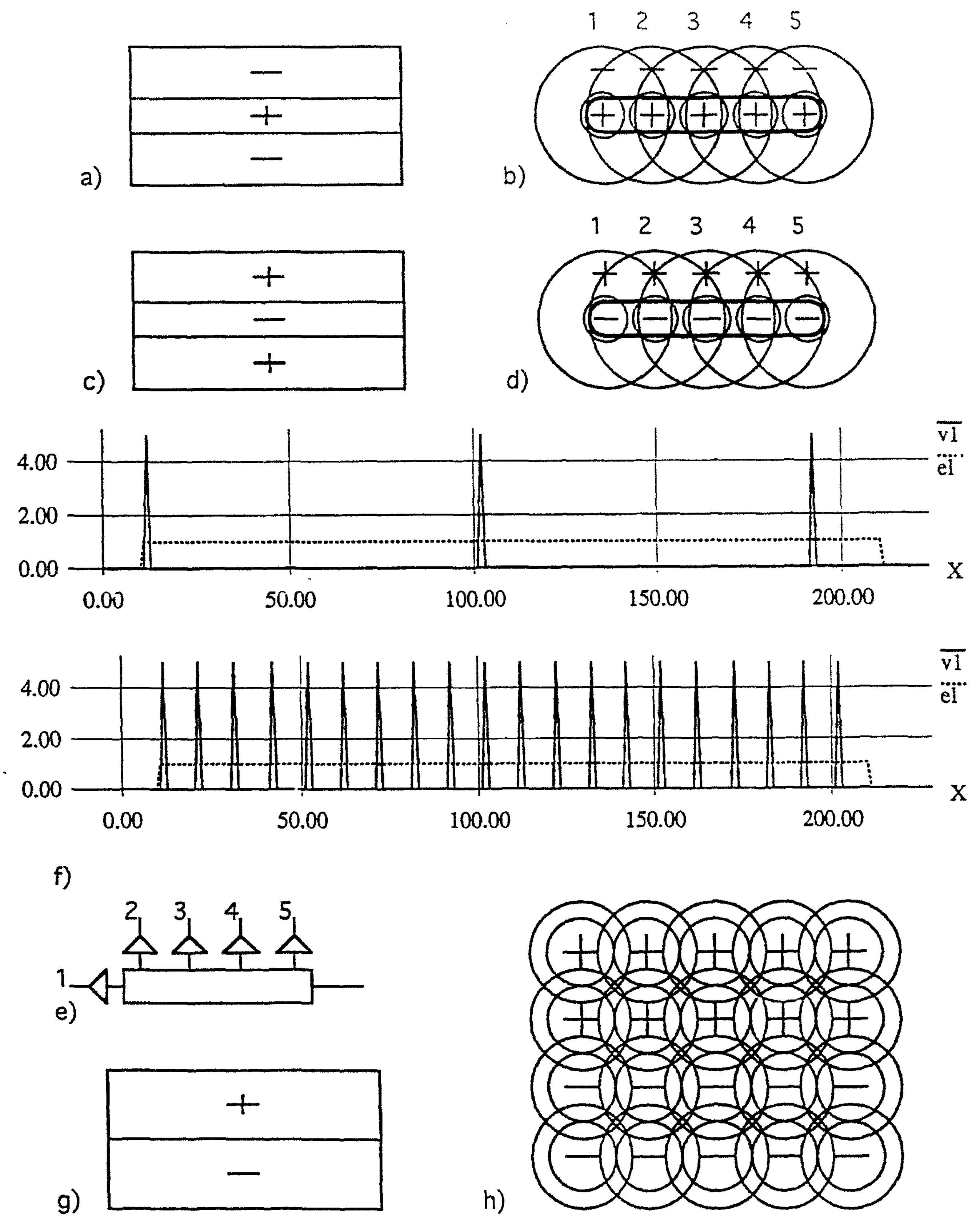


**Fig. 6:** Simple cell geometries. Three simple cell geometries and corresponding implementations; a) on-center line of light, b) implementation of (a), c) off-center line of light, d) implementation of (b), e) HTPE implementation of (a), f), simulated behavior of (e) - spot of light (top) and line of light (bottom), g) edge sensitive simple cell, and h) implementation of (g).

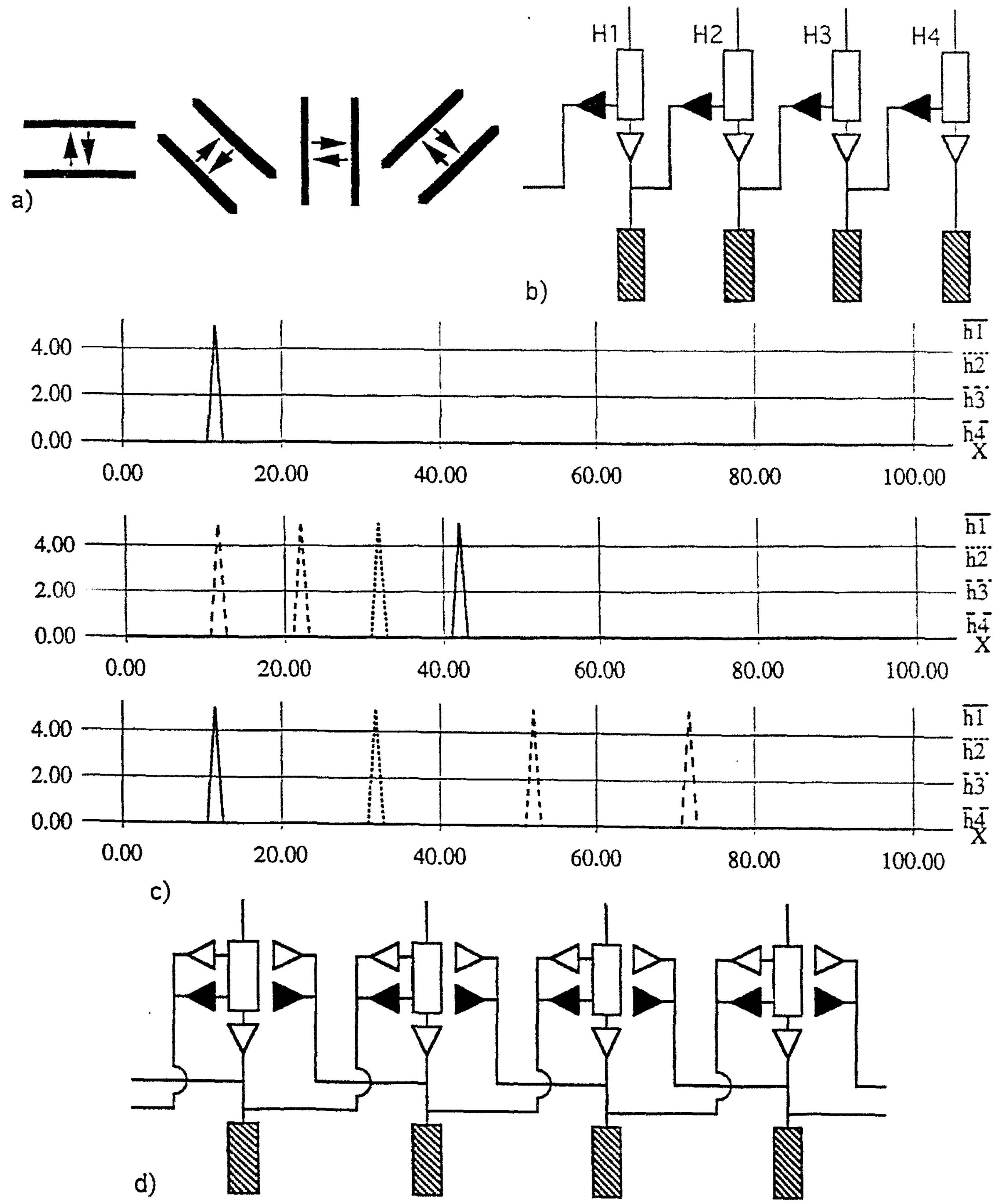
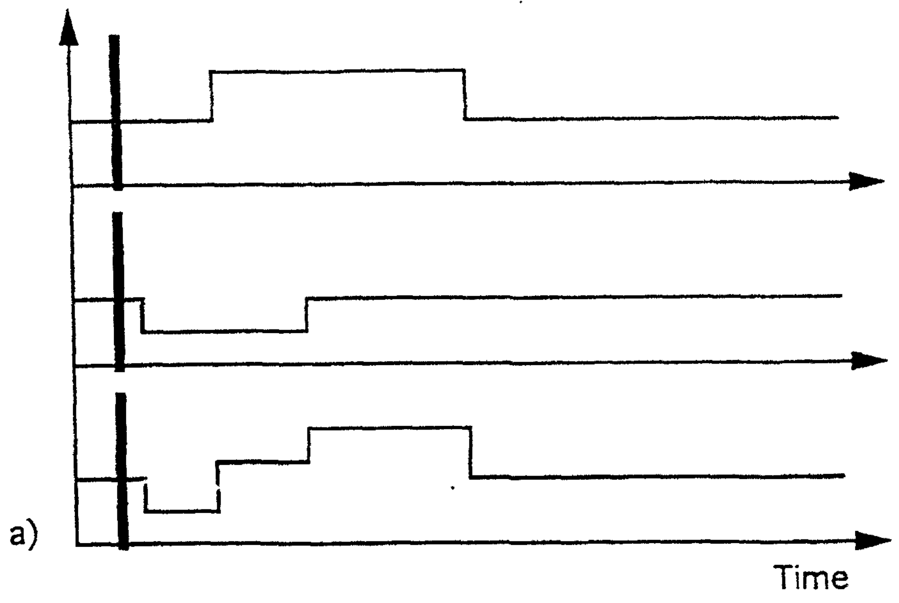


**Fig. 7:** Complex cell structures. HTPE-based complex cell structures and simulations; a) lines illustrating orientation and directional sensitivity of complex cells, b) HTPE complex cell network with right to left sensitivity, c) simulations of the network of (b); (top) response to fast left to right moving line, (center) response to fast right to left moving line, and (bottom) response to slow left to right moving line, and d) HTPE-based universal complex cell network.



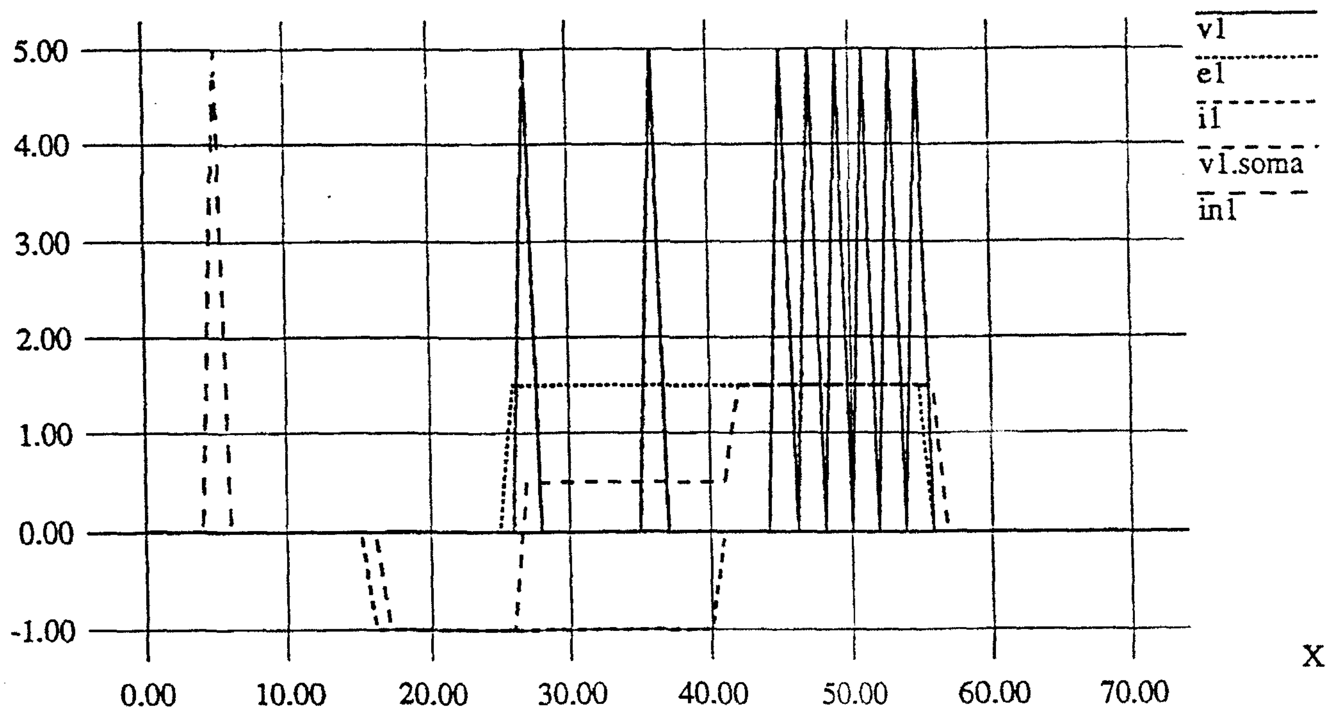
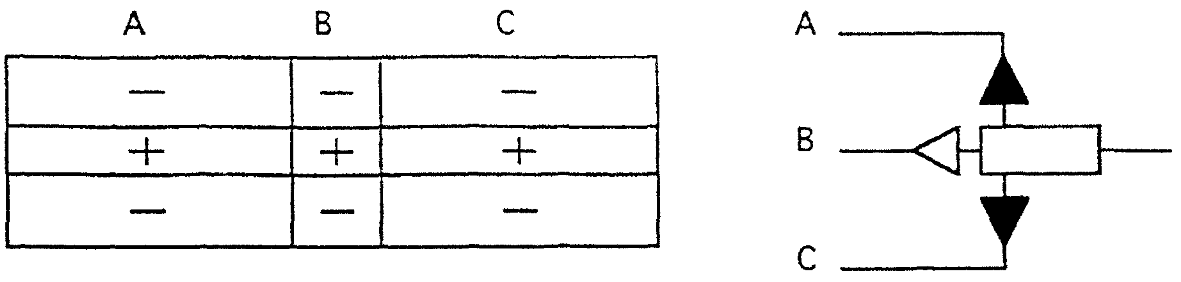
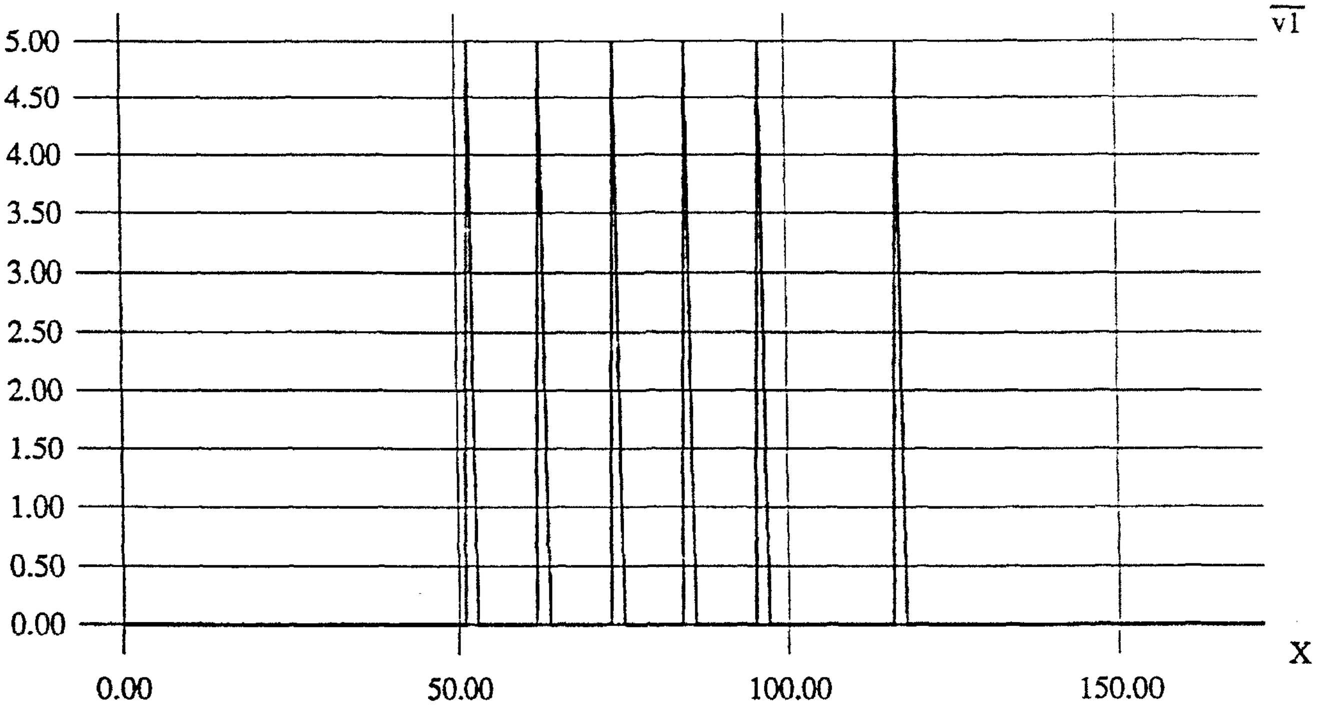


**Fig. 8:** HTPE waveforms. Time-dependent signals produced by HTPE; a) excitatory (top), inhibitory (center), and summed (bottom) waveforms, and b) simulation of HTPE with excitatory and inhibitory inputs, Note: HTPE firing rate is dependent on the net activity at its input.



a)



b)

**Fig. 9:** End-stopping complex cell. A possible implementation of an end-stopping complex cell and simulated results; a) multiple simple cells provide input to the complex cell, and b) simulation of complex cell as line of line stimulus grows causing the cell output to dissipate.

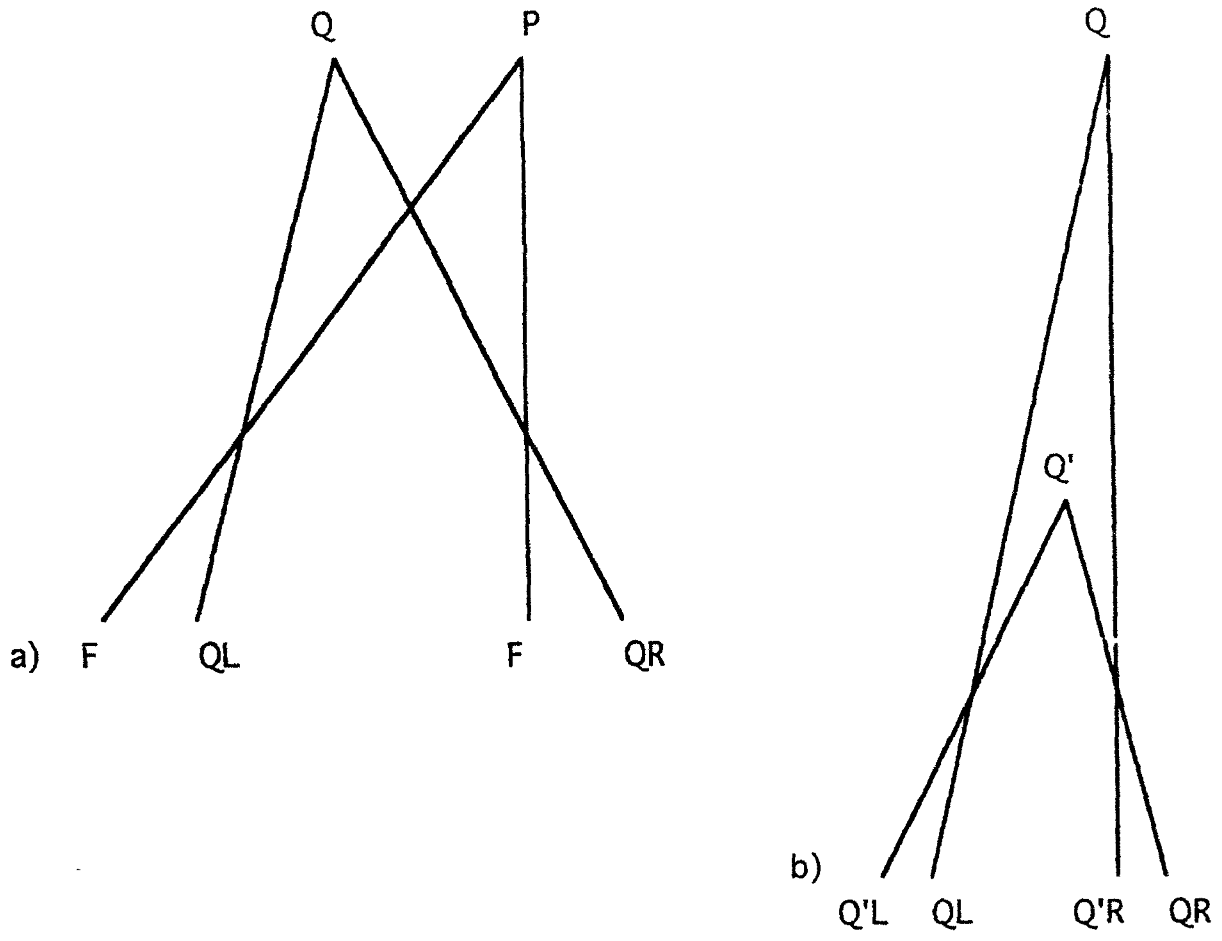


**Fig. 10:** Depth to disparity. Illustration of the depth to disparity mapping; a) no disparity, due to equidistanced objects P and Q, and b) disparity, due to distance difference between objects Q and Q'. The L and R represent left and right, respectively.